Product Configurators – Information System Support for Configurable Products[1]

Juha Tiihonen and Timo Soininen

Helsinki University of Technology
TAI Research Centre and Laboratory of Information Processing Science
Product Data Management Group
P.O. Box 9555, FIN-02015 HUT, Finland

Email: Juha.Tiihonen@hut.fi

Abstract. Product configurators have become an important information technology for the sales, order and delivery processes of many companies. In this article we discuss the types of configurable products and configuration related processes that can gain the most from such technology. Configurable products transfer much of the design work from the sales-delivery process to the R&D process. This requires systemizing the product and the related product knowledge. We present a general model of the R&D and sales-delivery processes for configurable products. The major benefits and problems of configurable products are then discussed.

This product and process-oriented view serves as a basis for our treatment of product configurators. Configurators with up-to-date product information allow non-product-experts to make error-free sales specifications and production orders. Configurators also reduce lead-times in the sales-delivery process. The major problem with product configurators are the long-term management and maintenance of the product knowledge as product models and product instances evolve.

The underlying thesis in this article is that product configurators on their own are not enough to make the sales and order fulfilment processes more efficient. The success of a configuration system in any company is based on adequate systemisation of the product, in some cases even re-designing the product for configurability, and the systemisation and reengineering of the configuration related processes.

---

[1] This article is also published in *Increasing Sales Productivity through the Use of Information Technology during the Sales Visit A Survey of the European Market*, Hewson Consulting Group, 1997.

1 Introduction

The design and production of goods that satisfy the specific needs of individual customers are of central interest to the European industry. The business environment of companies has changed in several respects. The major trends are diminishing lifetimes of products, increasing complexity and number of variants of products, and need for shorter lead-times in the sales-delivery processes. (We use the term *sales-delivery process* to describe all the phases required to propose, sell, order, manufacture and finally deliver an individual product to a customer.) Moreover, there is increasing pressure from the customers and the competitors to take individual customer requirements into account when specifying a customer-specific product instance.

One major trend to cope with these changes in the business environment is to improve customer-specific adaptation with *configurable products*. This type of product comprises a large number of variants and serves the specific needs of the individual customer by allowing customer-specific adaptation of the product. The goal is to do this while keeping the customer-specific adaptation easy, routine, and manageable, and the lead-time of the sales-delivery process short. In other words, a configurable product aims at combining the benefits of mass-produced and one-of-a-kind products. This type of operation has sometimes been called mass-customisation[2].

Configurable products are often very complex. *Product configurator* (or *configurator* for short) is an information system for managing products and their variants, and for doing customer-specific adaptation of the product. However, if a product configurator is to be of significant help in the sales-delivery process, the product also must be relatively easy to configure. Changing the business of the company to configurable products requires a strategic decision. The products and the related processes are affected. The required organisational and cultural changes can be very significant and difficult to achieve.

In this article we take a product and process-oriented view on product configuration tasks. The view is a synthesis of a survey of ten companies that deliver configurable products[3] and experiences gained in joint-projects with half a dozen other companies. The majority of these cases were from the discrete manufacturing industry, but similar thinking seems to be applicable to companies that produce services.

---

[2] See e.g. Hales H. L (ed.) Automating and Integrating the Sales Function: How to Profit From Complexity and Customization. *Enterprise Integration Strategies*, Vol. 9, no. 11, 1–9; Carson, C. (1997) Intelligent Sales Configuration. *PC AI*, January/February 1997.

[3] Tiihonen, J., Soininen, T., Männistö, T. and Sulonen, R. State of the practice in product configuration - a survey of 10 cases in the Finnish industry. *In Knowledge Intensive CAD, First Edition*, T. Tomiyama, M. Mäntylä, S. Finger, editors. Chapman & Hall 1996.

2 Configurable products

In this article the term *product* is used to mean the abstract specification or design of an entity that a company sells. Sometimes the terms 'product family' or 'product model' are used in the same sense. The product specification contains the information required by, for example, marketing, sales and production. The term *product instance* is used to mean a specific physical product that is to be delivered to a customer or a design of a physical product which is concrete enough to serve as a specification for manufacturing. In other words, a product instance is a customer-specific adaptation of the product.

A product consists of *components*. The term 'component' is used to mean an integrated whole which is considered a meaningful distinct part of the product. We use the terms 'component' and *component instance* in the same manner as the terms 'product' and 'product instance' to refer to a generic specification and a physical entity. A component may correspond to one or more physical parts of the product but may also be an integrated whole which is relevant for marketing purposes. The latter type of components are often called *modules*. A *pre-designed component* is a re-usable specification which either completely specifies a unique manufacturable or saleable entity or a clearly defined set of such alternative entities. The latter type of component is typically a parametric component. The distinction between products and components is somewhat ambiguous, because it depends on the point of view. For example, a product can sometimes be considered a component of another product. Some components may be sold separately, typically to be used as a spare part in or an extension of an existing product.

An *architecture* contains the information on the general arrangement of the product. The architecture can contain information on the structure, the required and optional components, and their topological or geometrical placements in the product.

2.1 Basic characteristics

A configurable product has the following basic properties:

- Each delivered product instance is tailored to the individual needs of an individual customer.

- The product has been pre-designed to meet a given range of different customer requirements.

- Each product instance is specified as a combination of pre-designed components or modules. Thus, there is no need to design new components as a part of the sales-delivery process.

- The product has a pre-designed general structure or architecture or a set of these.

- There is no need to do creative or innovative design as a part of the sales-delivery process. Rather, the specification of a product instance can be done in a routine manner.

The first property in the list differentiates configurable products from off-the-self, mass-produced products, which are not tailored for each customer. The second property reflects the fact that there is a clearly limited range of different customer requirements that the product is aimed at. In other words, the company does not intend to satisfy all the possible requirements with the product.

The rest of the properties differentiate configurable products from one-of-a-kind-products which are in principle designed from the scratch for each customer. Configurable products are pre-designed once as a part of a R&D (Research and Development) process, and then this advance design is used again and again in the sales-delivery process to specify product instances according to the requirements of each customer. In particular, one does not design new components or new ways of combining components to produce new functions that would satisfy any customer requirements, as is in principle possible for one-of-a-kind products (see the section Product development process produces configuration models).

Typical examples of configurable products include computers, lifts, trucks, telephone switches, modular furniture, heavy-duty diesel-engines, forest harvesters and dentist chairs. The products tend to be investment goods with corresponding price ranges, although there seems to be a trend to configuring also commodity goods, especially in the computer industry.

2.2 Product-related processes

The above properties of configurable products already hint at how configurable products are developed and subsequently sold and delivered. The product development process and sales-delivery process of configurable products are separate, like those of off-the-self, mass-produced products and unlike those of one-of-a-kind products.

When developing a new mass-product, the design process produces a specification of a single product which then is produced time and again in exactly the same form. For the one-of-a-kind product, the design in principle specifies a single product instance which is produced only one time, although some parts of the design may be re-used in other products. The product development process for a configurable product produces a *configuration model*, which explicitly defines the basic product properties and the possibilities for tailoring them (Figure 1). The configuration model is used again and again in the parts of the sales-delivery process related to configuration task, called

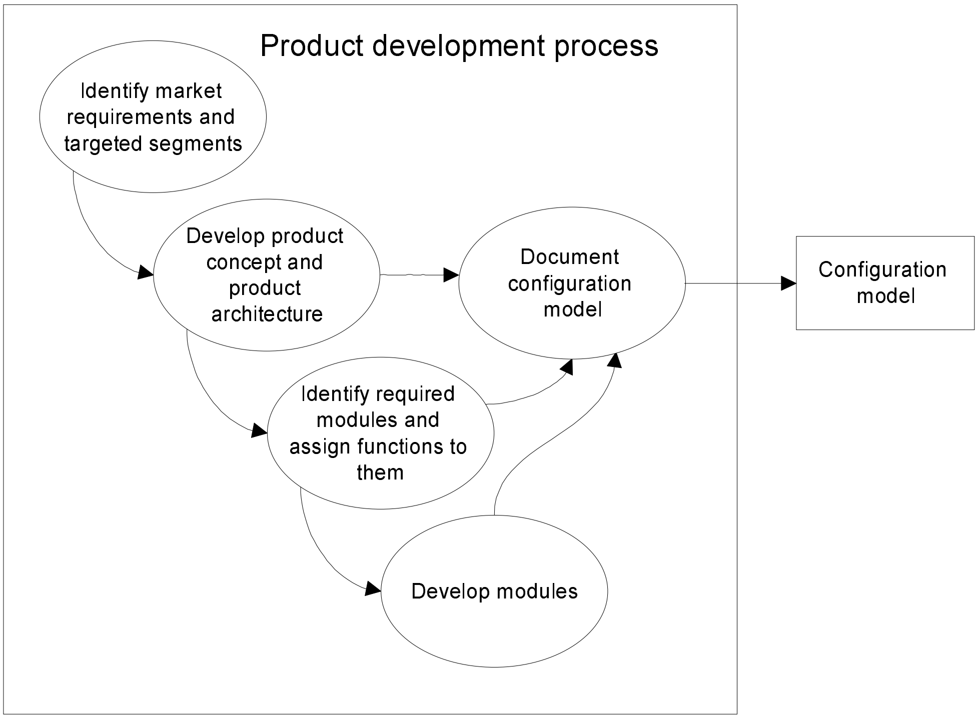

Figure 1. Product development process for configurable products

configuration process, as the source of information on the basis of which products are *configured* to produce customer-specific product instances. The task needed to produce a configuration is referred to as *configuration task*. A typical configurable product therefore enhances the re-use of design knowledge when compared to a similar one-of-a-kind product. Configuring a product on the basis of customer requirements produces a *configuration*, a description of the product instance to be delivered (Figure 2).

A configuration model contains all the required information on the possibilities of tailoring the configurable product according to the customer needs. In effect, a configuration model implicitly represents many, often millions, different products that could be treated as different mass-produced products.

A configuration model represents the valid combinations of the components that can be used to obtain the desired functions for the customer. The configuration model may contain information on the (possibly different) architecture(s) of the product, the available components from which the product may be combined, different interactions (constraints, rules, resource production-consumption relationships, etc.) between the components, the properties of the components and the customer needs that the components in different combinations satisfy. In practice there may be several different configuration models of the same product. These are typically meant to support the different phases of the configuration process, for example sales and engineering configuration tasks, at a suitable level of abstraction. The sales function typically does not need precise definitions required for generating, for

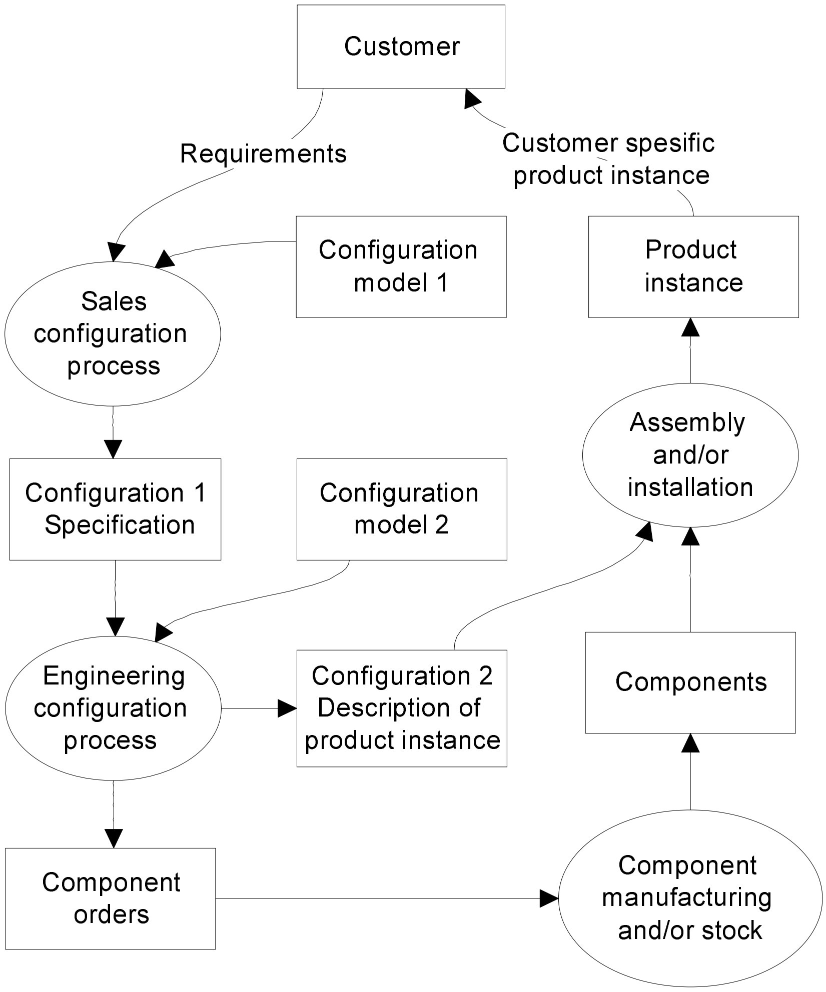

Figure 2. Sales-delivery process of configurable products

example, a detailed Bill-of-Material (BOM) of the product as a part of the configuration. This information is needed in the engineering configuration tasks for configuring the product. The sales persons may prefer to view the product as a set of alternative and optional functions that it may produce or as a set of alternative modules.

2.2.1 Product development process produces configuration models

The role of the product development process is to develop the product and its configuration model (Figure 1). After market segments and their requirements have been identified, one must decide to which segments the product and each of its modules are targeted. Matching the product architecture and modules to typical combinations of requirements is critical. A mismatch either nullifies the sales opportunities or leads to excessive one-of-a-kind design, because suitable alternatives are not available just by configuring the product.

A configurable product concept is typically implemented with a modular product architecture because integral product architectures do not facilitate configuring and evolution of the product[4].

It may be difficult to decide which alternatives should be included in one product and which should be treated as separate products ("Should we have 50 simple or 5 more complex products?"). Quite often, less complex models are favoured due to difficulties in the maintenance of complex models and due to the possible difficulties in communicating the possibilities of a very flexible product to customers.

It is possible to extend a configurable product by adding new modules or replacing old ones to meet new or changed customer requirements. This may enable longer product life cycles. This option should not be used as a poor substitute for finding out the real requirements of the market. The reasons are numerous, but they include: 1) lost sales due to a longer time required to develop a viable product and due to inappropriate offerings while the product is being enhanced 2) late fixes or poor understanding of requirements that lead to a sub-optimal product that covers the real requirements only partially or requires too costly or otherwise ineffective combinations of modules to meet the real requirements.

The product should be easy to configure. For example, selection of one module should ideally not affect the selection of other modules. Selling a complicated product is error-prone and may be beyond the technical capabilities of sales persons. Although configurators can handle even very complex module interactions, the maintenance of such configuration models may become prohibitively difficult. Good results in using a configurator can be expected only when the product has been designed for easy configurability.

Documenting the product configuration model systematically is a major task, but it is vital for both manual configuration tasks and successful use of a configurator. Decisions on the contents of the configuration model should be done early in the product life-cycle. In an ideal case a product expert with a clear understanding of the entire product enters the model to the configurator. If a computer expert enters the configuration model, he/she should not have to invent the configuration model or gather the required information from multiple sources. Product policy and commercial and technical restrictions of the product are typically clearly out of scope of expertise of these persons.

Many companies want to serve all the customers and deliver products that are tailored beyond the possibilities defined in the configuration model. Usually the required modifications concern only one or two modules. Quite often in these cases most of the product can be configured and only the customer-specific variations need to be designed. It may be more cost-effective to engineer rare variants case by case rather than try to extend the product to cover all the possible requirements.

[4] Ulrich, K. and Eppinger, S. *Product Design and Development.* McGraw-Hill, Inc., 1995.

2.2.2 Delivery process produces a configuration

A request-for-proposal or an order from the customer initiates the configuration process (Figure 2). The inputs of the process are the customer requirements, which may be incrementally defined and refined as the process advances, and the configuration models. The configuration process produces a configuration on the basis of these. Eventually, the configuration should contain the necessary information for the production to manufacture and logistics process to deliver and install the product instance, for example a BOM or a layout drawing.

As shown in Figure 2, the process may contain several stages, such as sales and logistics-centre or engineering configuration. In the sales stage the product instance may be specified in terms of the abstract functions or modules that satisfy the customer requirements, which results in a sales specification or order (configuration 1). In the engineering stage the output of the preceding stage is used in connection with the configuration model for this stage to produce a more concrete definition of the product instance (configuration 2). The configuration process in a company may not contain all these stages, for example all of the configuration task may be done in the sales or engineering stage.

The sub-tasks of the configuration task may include:

- finding out and recording customer requirements;
- product architecture selection;
- component selection;
- parameter value determination for component instances;
- layout design;
- determination of component instance connections;
- determination of the price of the product instance based on the configuration;
- determination of the delivery time;
- preparing a proposal;
- preparing a technical specification;
- completeness checks;
- consistency checks.

Configuration process typically includes at least finding out the customer requirements, component selection, determination of the price of the product instance, preparing a proposal and a technical specification, and checking the completeness and consistency of the configuration. The complexity of the configuration process increases if some of the remaining tasks are included, especially in the case of layout and component connection design. In some cases it may be necessary to generate several alternative configurations that satisfy the customer requirements. In these cases the most appropriate one must also be chosen on the basis of some optimality criteria as a part of the configuration process.

2.3 Long term management of configuration models and configurations

Products evolve as customer requirements change. New products are developed, new functions and possibilities of variation are added to an existing product, and sometimes functions and possibilities are removed because they are not in demand. This means that the configuration models must evolve accordingly. If the pace of change is high, the management of configuration models may become a problem.

In addition to managing the configuration model, the companies often need to record and keep up-to-date the information on the product instances as they are serviced ("as-built"- and "as-maintained"-configurations) and possibly modernised. The latter type of changes to the configuration, *reconfiguring*, is sometimes so important to a company that it systemizes the knowledge needed for reconfiguration tasks as special configuration models which contain the information on how new components can be fitted to an existing product instance to achieve new functions. These models may be very complex because they bridge the gap between temporally different versions of configuration models and model the interdependencies between different versions of components.

2.4 Some advantages and problems

The primary advantages of configurable products are[5]:

- The ability to fulfil a wide range of customer requirements.
- Shorter delivery times and smaller stock.
- Increased control of production, for example, a large variety of end products with a fairly small number of components.

[5] Tiihonen, et al. State of the practice in product configuration.

The sales-delivery process of configurable products has a shorter lead-time than one-of-a-kind products. Several companies that have previously delivered one-of-a-kind products have moved to configurable products to cut down the delivery times and to re-use the product knowledge generated in product development[6] (Figure 3). When making this transition the main effort probably goes to pre-designing and systemizing the products so that they are configurable. This may require considerable investments in product development, modularising the products and systemizing the product knowledge which often only exists in the product developers' minds. In addition, the sales force must learn not to offer unnecessary changes to the product which would result in customer-specific design. This cultural change in the process may be difficult to accept. The investment is profitable only if the volume of delivered product instances is high enough. The volume needed is of course very much dependent on the market situation and the type of industry. It seems that a volume of hundreds of delivered product instances is enough to justify the investment[7]. It is often not feasible to offer only configurable products. In addition, one-of-a-kind products are offered for those customers that need the special functions and are willing to pay more for them than for configurable products.

A company may move towards configurable products from the opposite direction (Figure 3). Companies that have previously offered mass-produced products have also moved to configurable products[8]. This has usually happened to address the need for a wider selection of products than can be efficiently managed as fixed, mass-produced products. However, it seems that it is not feasible to change low-priced, really high-volume commodity products to configurable products. When moving from mass products to configurable products the types of functions that the customers require and the ways to combine the components to produce them may be better known than in the case of one-of-a-kind products. However, introducing a configuration process to the delivery process which has previously operated with fixed mass products can cause problems, as more of the specification work is expected to be done by the sales persons and in the engineering configuration. All the processes in the company and the supporting systems must be altered to handle customer-specific variants of products.

In both one-of-a-kind- and mass-production oriented processes the necessary changes may be difficult to achieve. Producing and maintaining the configuration models and configuring the product instances both require systemizing the processes and documenting the relevant product knowledge. Without systematic documentation of the configuration models the sales persons do not have a clear understanding of the possibilities of the product and specify erroneous configurations due to lack of knowledge. Even if the configuration models are systematically documented but the sales persons do

---

[6] *ibid.*

[7] *ibid.*

[8] *ibid.*

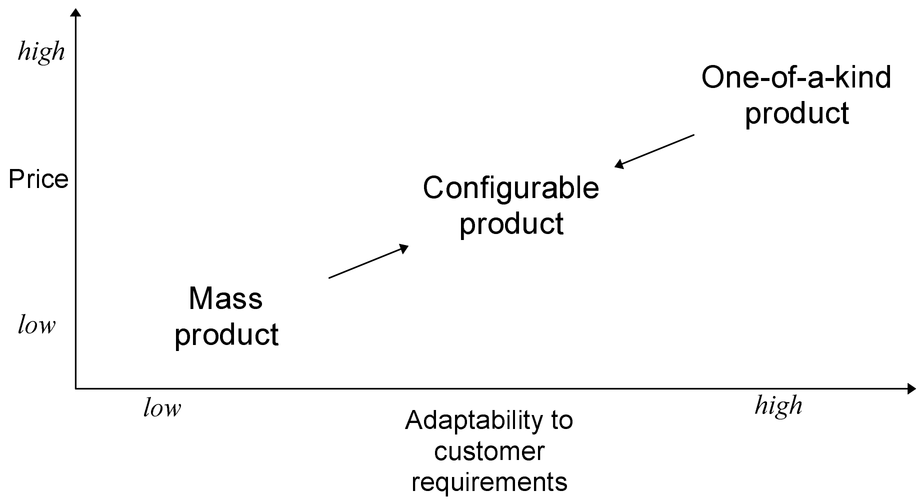

Figure 3. Configurable products compared to mass- and one-of-a-kind products

not systematically use them, the orders will contain errors which cause iteration and problems in the latter stages of the delivery process. The importance of a systematic, error-free and efficient operation of the first stages in the sales-delivery process is increased when moving to configurable products. It is necessary that the product knowledge of the configurers is correct. Support for configuration tasks becomes important.

The problems in the development of the processes are to large extent due to the fact that the increased effort and the benefits gained are experienced by different functions within the company. The benefits of a systematic configuration model produced by the product development department are felt by the sales department. The benefits of systematic configuration process in the sales are gained by the later stages in the process. It is sometimes a problem to create a company-wide understanding of the benefits of systematic processes and systematic documentation. It may be that in a more process-oriented organisation the required change is easier to implement than in a functional organisation. Usually a process-oriented organisation increases the understanding of the participants on the needs of other participants in the process.

3 Product Configurators

A product configurator is an information system that is used to configure product instances and to create and manage configuration models of products. Product configurators have been used as an aid in the sales-delivery process at least from the beginning of the 1980's[9]. There are numerous theoretical models of product configuration tasks[10] and reports on implemented systems that support product configuration tasks[11]. There are at the least a few dozen commercial systems available[12], although the situation changes so rapidly that it is hard to give an exact figure.

3.1 Motivation for using a product configurator

There are several reasons for using a product configurator as a support tool for the sales-delivery process. The fundamental ones are the complexity of the product, and the ability to reengineer the business processes through using a configurator as an essential enabler[13]. Other reasons that can be traced to the two fundamental ones include needs to[14]:

- cut down the number of errors in the configuration process,
- reduce the lead-times in the sales-delivery process,
- efficiently distribute up-to-date product knowledge to the people in the configuration process, and,
- increase the volume of quotations without increasing the number of sales persons and engineers.

---

[9] A classical example is documented in: McDermott J. R1: a rule-based configurer of computer systems. In *Artificial Intelligence*, vol. 19, no 1, 39–88, 1982.

[10] See e.g.: Mittal S., Frayman F. Towards a Generic Model of Configuration Tasks. In *IJCAI-89: proceedings of the Eleventh IJCAI*, 1395-1401, 1989; Heinrich M., and Jungst E. A Resource-Based Paradigm for the Configuring of Technical Systems from Modular Components. *In Seventh IEEE Conference on Artificial Intelligence Applications, [proceedings]*, 257–64, 1991; Najman O., Stein B. A Theoretical Framework for Configurations. In *Proceedings of Industrial and engineering applications of artificial intelligence and expert systems : 5th international conference, IEA/AIE - 92*, Belli F., Radermacher F. J. (eds.), 441–50, 1992.

[11] See, for example: Hales H. L Automating and Integrating the Sales Function: How to Profit From Complexity and Customization; Cunis R., Günter A., Syska I., Peters H., Bode H. PLAKON–An Approach to Domain-Independent Construction. In *The Second International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems IEA/AIE –89: proceedings*, 866–74, 1989; McDermott J. R1: a rule-based configurer of computer systems.

[12] The configuration homepage at http://www.cs.unh.edu/ccc/config/ provides a list of references to the homepages of vendors of commercial systems under the heading Who's Who/Companies. See also the vendor information in this report.

[13] Hammer, M. & Champy, J. *Re-engineering the Corporation - A Manifesto for Business Revolution.* HarperCollins, New York, USA, 1993.

[14] Tiihonen, et al. State of the practice in product configuration.

A product may be too complex for a sales-person without specific engineering skills and technical background to configure. In this case the configuration task (without a product configurator) is often done in co-operation between a sales-person and an engineer. This leads to the need of iterating between the customer, sales person and engineer to obtain a working configuration that satisfies the customer needs (thick lines in Figure 4). This in turn leads to longer lead-times, to several possibilities of misunderstanding the actual customer requirements, and to more opportunities for making mistakes, as the people who actually configure the product instance are not the ones that communicate with the customer.

A commonly cited figure is that the companies lose 1-2 per cent of their annual revenue due to errors in the sales configuration tasks[15]. Another study revealed that companies lose 10-15 per cent of their annual turnover through inefficiency in their sales-delivery processes[16]. The latter inefficiency is not entirely due to configuration task related problems. However, the errors are usually made in the early stages of the sales-delivery process, particularly in the sales and to some extent also in engineering configuration[17]. It is very typical that about 80 % of the sales configurations are incomplete and about 50 % of them contain errors[18]. These errors cost a lot to correct if they are only found during the manufacturing or delivery and installation of the product instance. In this case the quality and efficiency of the rest of the sales-delivery process is of no help, as the later stages rely on erroneous information. The manufacturing and logistics processes have often been rationalised and automated, and function very efficiently, but the sales and R&D-processes have not been developed as much[19]. There is usually a greater possibility of noticeable improvement in the sales process than in the production process when investing in the quality and efficiency of the processes. In a textbook example from the Finnish industry, the lead-time of the configuration process was eight weeks and that of manufacturing was only two weeks.

The sales-delivery process can be reengineered by giving the sales person the ability to configure a complex product without the need for consulting an engineer. This makes it possible for the sales-person to explore different possibilities for satisfying the customer requirements interactively with the customer. The sales-person can also concentrate on selling the product and not on the technical details.

---

15 See, e.g.: PC AI, January/February 1996. McHugh, J. "Holy cow, no one's done this!" *Forbes*, June 3, 1996.

16 Luhtala M., Kilpinen E. and Anttila P. *Logi : managing make-to-order supply chains*. Report/Helsinki University of Technology, Industrial Economics and Industrial Psychology 153, (1994).

17 Tiihonen, et al. State of the practice in product configuration.

18 *ibid.*

19 McHugh, J. "Holy cow, no one's done this!".

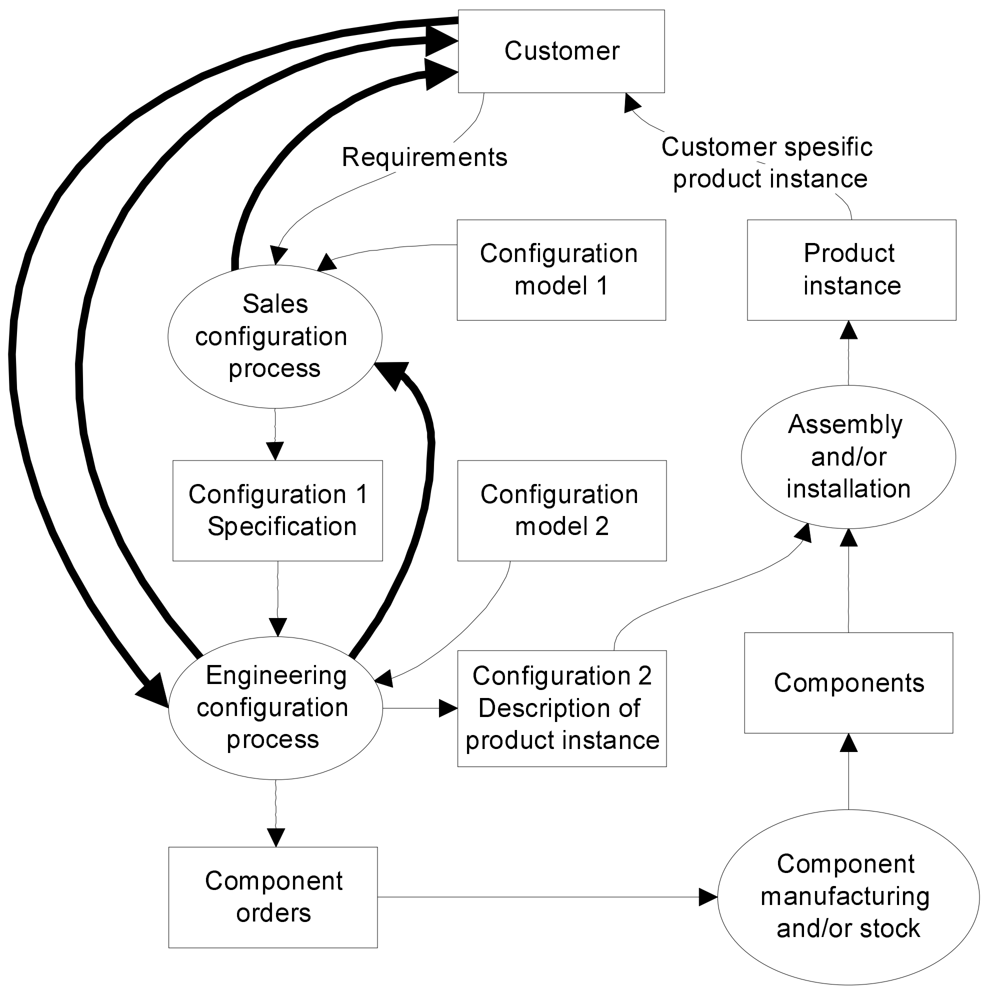

Figure 4. A typical sales-delivery process without configurator support

The iteration between customers, sales persons and engineers can be eliminated with a product configurator (Figure 5). The main effect on lead-times comes from this stream-lining of the configuration process. The difference in the lead-time between an iterative process and a configurator enabled process can easily be tenfold. The ease of configuring a complex product also reduces the time required to generate a configuration. The integration of a configurator to MRP (Manufacturing Resource Planning) or ERP (Enterprise Resource Planning) and PDM (Product Data Management) systems also reduces the lead-times, as the configuration can be available to the later stages of the sales-delivery process immediately upon the confirmation of the order.

The use of a product configurator guarantees that the configurations are always error-free and complete with respect to the configuration model, and that the order is immediately available to the latter stages of the process. This is an advantage also when the product is simple enough for a sales person without engineering skills to configure without a configurator. However, the correctness of the configurations is dependent on the correctness and

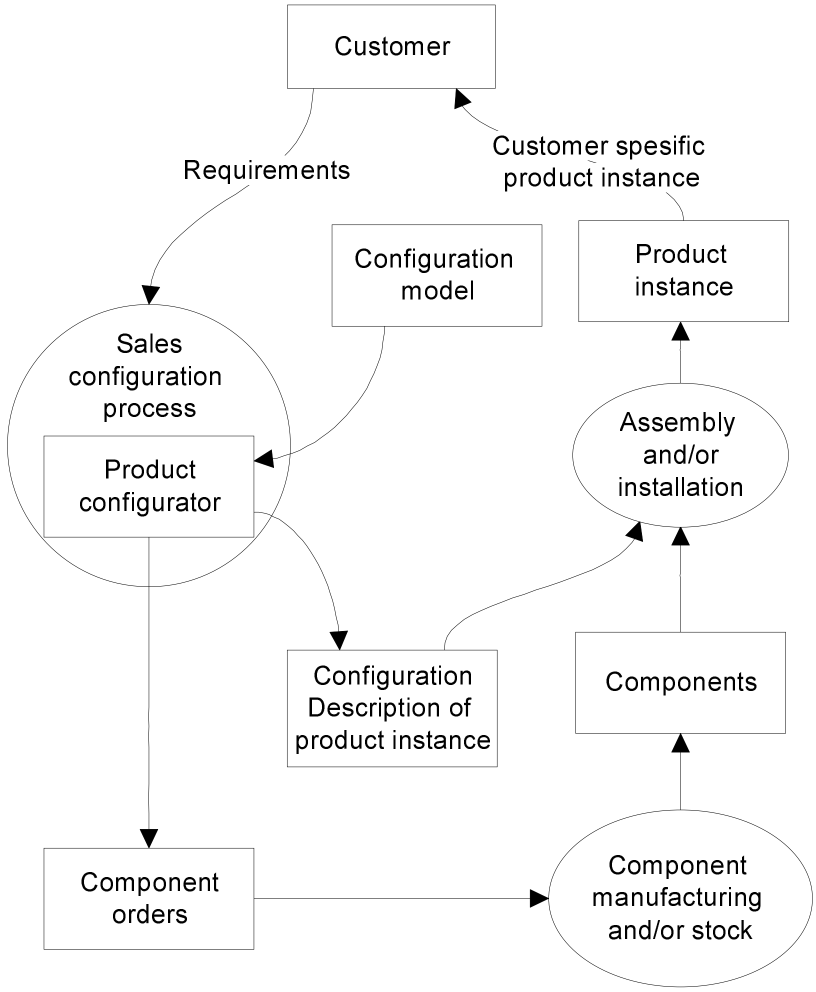

Figure 5. A reengineered sales-delivery process

completeness of the configuration model. These properties of the configuration model cannot be ensured by configurators. Validating the correctness and completeness of the configuration model is especially difficult for complex products. Another source of errors which cannot be eliminated by using a product configurator is that the sales person may misunderstand the requirements of the customer.

A product configurator also functions as a media through which new product knowledge, particularly the configuration models, can be distributed to the people in the configuration process. Unlike other media, a configurator is also a means to enforce the use of the up-to-date knowledge when configuring the product.

3.2 Simple classifications of product configurators

Product configurators can be roughly categorised as single-use and general use systems, and on the other hand, as academic prototypes and commercial systems. A single use-system is designed to aid in the sales-delivery process of a particular product, or a set of products, of one company. General use systems can be used to model and configure different types of products in different companies.

A single-use configurator is usually programmed in some suitable programming language, while the general use systems try to minimise the amount of programming needed for an individual configuration model. This is done by representing the configuration model in a suitable modelling language which directly supports the configuration domain. The systems in the 1980's were largely either single-use commercial or general use academic prototypes. Today, starting roughly from the beginning of this decade, there is a rapidly increasing number of commercial general use systems available, which indicates that the technology has reached a certain maturity level and acceptance. However, not all the problems have been solved.

3.2.1 Primitive, interactive and automatic

Product configurators can also be classified on the basis of the functions that they provide to the person doing the configuration task. At the simplest level, the system merely records the configuration decisions made by the user. The system does not check that the decisions are valid with respect to each other, or that all the necessary choices have been made. This type of systems may be classified as support tools for ordering, but can also be considered *primitive configurators*. At a more advanced level, an *interactive configurator* can check the validity of the configuration decisions made by the user in addition to recording them, and it can also guide the user to making all the necessary decisions. This checking can be done after each user decision, or it may be done at certain check-points, e.g., after the user has made all the decision. Full support for configuration tasks is provided by an *automatic configurator*. In addition to the functionality of an interactive configurator, an automatic configurator can, on the basis of some customer requirements, generate parts of or even the entire configuration automatically.

Automatic configurators may differ in their capability to make "intelligent" configuration decisions. Some may only make simple decisions on the basis of the user choices, while others can make all the necessary decisions to produce a configuration. The commercially available configurators are usually automatic with at least some intelligent capability, but for some companies an interactive configurator may be all that is needed.

3.2.2 Structure-, resource- and logic-oriented approaches

There is no unique widely accepted way of modelling the product or configuring the product instance on the basis of the configuration model. Therefore the product configurators are based on different principles and ideas. These include structure-oriented approaches, resource-oriented approaches, and logic-oriented approaches (constraints and logical sentences).

A structure-oriented configurator models a product by the structures or physical connections of the components. A resource oriented configurator models a product by specifying the types of resources different components produce and consume. A logic oriented configurator specifies the interactions of the components as constraints or logical rules. A commercial configurator often combines some of these approaches to ease modelling the different types of knowledge in a configuration model.

3.3 Problems with product configurators

In addition to the general problems of any software meant for automating the processes within a company, the currently available commercial configurators have two specific problems. The most difficult problem is the long-term management of the product knowledge in the configuration models and the configurations. The second is the diversity of modelling approaches and functions provided by the systems of different vendors, which makes it difficult for a company to decide which particular system best matches its needs.

In addition, deciding the functions that are desired from a configurator can be difficult. Further, the integration of configurators to other systems may also be problematic.

3.3.1 Long-term management of configuration models

The long term management of configuration models has been a significant problem for single-use configurators. A classical example is the XCON-configurator at Digital. Some 40 % of its program of tens of thousands of rules had to be rewritten every year. The maintenance of the configurator needed in excess of 25 people, even though it had been programmed in a rather high-level rule language[20]. A single-use configurator may in principle be programmed just like any software. Both the configuration model and the functionality of the configurator are then programmed in a programming language. Usually they become mixed with each other. As the logic of the program and the description of the product are mixed together, the maintenance of the system is very difficult. Further, representing a configuration

---

[20] Barker, V.E. and O'Connor, D.E. Expert systems for configuration at Digital: XCON and beyond. *Communications of the ACM*, Vol. 32, no. 3, 298–318, 1989.

model in a traditional programming language, which is not designed for the configuration domain, is difficult and results in large and complex programs. A general use system clearly separates the *configuration engine*, which contains the logic and functionality of the configurator, from the configuration model of a product. The configuration model is represented in a domain-specific (and configuration engine specific) language, which results in compact and easy to understand models.

There are several examples that show that the maintenance of a single-use configurator has been extremely difficult. The information in the configurator could not be kept up to date which led to the users having to check the configuration manually. In an extreme case the use of the configurator became impossible. In the worst case the non-maintainability of a configurator may even delay the introduction of new products and options, which can be disastrous in today's competitive situation.

A general use configurator is the only viable alternative because configuration models must evolve at the same pace as the products evolve. This pace of changes is increasing, which means that the long term management problem is getting worse. A general use configurator allows one to directly to model the configuration model with minimal programming effort. This is a pre-requisite for successful long-term management of product configurators, but it may not be enough in the long-run. In addition to it, a configurator should have the following properties:

- an intuitive, understandable, visual way of modelling;
- a structured, probably object-oriented, method of modelling;
- means for modelling directly the evolution of products, components and their interdependencies.

The first property aims to ensure that the configuration models can be managed by the product developers, product managers and other people in the company who understand the products best. This eliminates the error-prone and lengthy knowledge acquisition of product knowledge. There is no need to have computer experts that "code" the information, which spreads the management effort to more people. The second property increases the understandability of the models and consequently helps managing the model by dividing it into relatively independent packages of knowledge whose interactions are well-defined. Object-oriented modelling is one popular way of realising these benefits. The third property adds to the modelling method some aspects of configuration management[21] and product data management (PDM)[22]. These may include versions of products, versions of components

---

[21] Buckley, F. *Configuration Management. Hardware, Software, and Firmware.* IEEE Press, 1993.

[22] *PDM Buyer's Guide Product Data Management Systems for Improving Processes and Products,* Sixth Edition, Volumes 1 and 2. CIMdata, Inc., 1997.

and versions of dependencies between components and products, effectivity intervals, and dependencies between the different versions.

Today's commercially available systems have the first two properties with varying degrees but there is no generally accepted good way of achieving them. The third property may be found in some systems that are tightly integrated with PDM systems, but this is much rarer. Therefore the capability of configurators to manage the configuration models of a given company can be judged by how easy and natural it is to model the products of the company and by whether the product experts can easily understand and manage the models. The applicability of a commercial system is affected by the expected pace of changes to the products.

A considerable problem related to the management of configuration models has also been the effective distribution of the configurator to the entire sales force. It may be necessary to support both automated and manual configuration tasks if all retailers (for example, those with low sales volumes) are not ready to use such technology. This problem is worse if a retailer sells the products of several manufacturers and should operate several configurators. The solution to this problem seems to be to centralise the support for configuration tasks and to implement a web-browser-interface to the centralised configurator. As always, when operating in the Internet, there are problems with security in this approach but these can probably be solved with appropriate security technology.

3.3.2 Long-term management of configurations

The long term management of configurations is at its most complex when there is a need for modelling the knowledge on reconfiguration. A pre-requisite for supporting re-configuration tasks over generations of products is the ability to model versions and effectivity intervals and to store the changes to the configurations of the product instances. Given this information, an engineer can check what the customer's product instance contains and check which component instances may have to be changed. The actual reconfiguration tasks must be done by the engineer, however. This type of support is mostly lacking in the present configurators, which do not support temporal modelling. Research is still needed on more advanced support that would automatically or semi-automatically compute the required changes to an existing configuration over several versions of products and components[23].

[23] Männistö, T., Peltonen, H. and Sulonen, R. View to product configuration knowledge modelling and evolution. In Boi Faltings and Eugene Freuder, editors, *Configuration—Papers from the 1996 AAAI Fall Symposium*, pages 111–118. AAAI Press, 1996. The American Association for Artificial Intelligence, Technical Report FS-96-03.

3.3.3 Choosing a configurator

Here we concentrate only on the configurator specific problems in choosing a suitable configurator and leave the general questions on choosing software aside. These include the capability and willingness of the vendor to produce what is promised with an acceptable schedule, the level of support that can be expected from the vendor after buying the software, the platforms that the software runs on, etc.

The lack of a unique widely accepted way to model and configure a product is a basic problem in choosing a commercial configurator. A company must have a clear understanding of its needs before it can choose the configurator whose principles and functionality best suit them. This enables compact and easy-to-maintain configuration models. Particularly, little programming effort should be needed to get the desired functionality. It is of utmost importance to realise that it is not enough that a configurator can model and configure the product. This should be done with minimum programming effort, and with as little modelling effort as possible. This is the major pre-requisite for the maintainability of the system.

As a product configurator affects the whole sales-delivery process and a part of the R&D-process, the company must understand the needs of all the functions taking part in the process and all the implications of introducing the new technology to the processes. Coupled with the often present need for business process reengineering, the task of choosing a configurator usually takes more resources and time than expected. A particularly difficult task is to ensure that the sales function takes actively part in defining the needs for the configurator. The sales persons are often not used to taking part in defining complicated operative systems, unlike the production engineers, for example.

Another serious problem for small to medium sized companies (in European standards) has been that the investment in the software, licenses, consultation, modelling the products and then maintaining the models can be quite high. However, there are more and more vendors in the configurator market and the competition is getting tougher. There are some configurators in a suitable price range for a medium-sized company and there will probably be affordable solutions for small companies within a few years.

3.3.4 Other considerations

Manual configuration skills may deteriorate when an automated configurator is used for an extended period. Loss of personal proficiency may be frustrating for skilled employees. Due to this, some companies do not want to have automatic configurators. These effects may occur if the employee is only expected to do routine configuration tasks using the configurator. However, if the employees can handle more of the more complex cases (that cannot be handled by a configurator) then the knowledge and motivation of the employees will probably be preserved and may even increase, as the employees are not tied down to routine tasks. In any case, the increased time for discussions and selling-oriented work made possible by the configurator will probably increase the motivation of the sales persons. In order to keep the product knowledge of the sales persons up to date, the configurator should be able to explain the reasons behind automatically made decisions[24].

Another reason for avoiding automatic configurators is that many companies seem to have a deep distrust on artificial intelligence systems. Users want to retain control of the configuration process, although they want to have advanced support. For these reasons, interactive configurators are often favoured.

In many cases, the sales view of the product is simpler and more function oriented than that of engineering. This should be reflected in the configuration model. To support engineering configuration tasks, deeper knowledge (i.e. knowledge with more explanatory power) is often required. The model is usually more structure oriented and more complicated: instead of product functions, the interest is centred on the components, their properties and relationships.

Configurators need to be interfaced or integrated primarily to ERP systems, PDM systems and CAD systems. Usually integration must be dealt with case by case, as there are no common interfaces. Thus, interfacing may represent significant costs. Some configurators include interfaces to leading ERP systems.

Integration to an ERP system is required for two reasons: The primary purpose is to submit customer-specific orders to the company's operative system. In addition, delivery time and capacity information from the operative system may be used by the configurator.

Integration to PDM may be required for acquiring configuration model information from the PDM system and for storing configuration models in the PDM system. In addition, configurations may be stored in the PDM to serve as a basis for as-maintained configurations.

Integration to CAD may be required if advanced layout or geometric design is needed during the configuration task. Some CAD systems can export part lists of assemblies, but in our view this is generally not among the primary integration needs.

4 Conclusions

Configurable products and product configurators are becoming an important issue in the industry. However, a company should carefully consider the risks in offering only configurable products. A mixture of configurable and mass-produced or one-of-a-kind products can often more accurately correspond to the needs of the customers. It is also often possible to gain some of the benefits of configurable products by developing mostly configurable products,

[24] This effect of the explanation-function was brought to our attention by Klas Orsvärn.

whose customer-specific adaptation can be done by designing only very restricted parts of the product for each customer.

A product configurator can be a major reengineering asset for a company. The benefits of using such a system may be truly dramatic: zero percent errors in orders, tenfold improvements in the lead-times and better management of the product. In order to reach these goals, a company must systemize its products, processes and knowledge. Acquiring a product configurator is therefore usually not only an information system project. Rather, efficient use of the configurator usually requires also fundamental reengineering and development of the processes of the company.

The pit-fall of product configurators has been and still is the long-term management of product information. This item is especially important for the successful use of a configurator in a company. Choosing between the different systems in the market is not easy, as the different systems operate on different principles and ideas. The long-term management should be a major criterion for choosing the system.

Acknowledgements

The work reported in this paper has been partly funded by the Technology Development Centre of Finland and the Academy of Finland (project 29655). We thank Hannu Peltonen and Klas Orsvärn for reviewing this paper and for their excellent comments on how to improve it.